Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
д-р техн. наук, профессор
_____ А.В. Замятин
подпись
«_23_» _мая_____ 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА ШЛЮЗА К CLICKHOUSE В СИСТЕМЕ ОБРАБОТКИ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТОВ A/B-ТЕСТИРОВАНИЯ

по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии,
направленность (профиль) «02.03.02 Фундаментальная информатика и информационные технологии»

Юдаков Алексей Александрович

Руководитель ВКР
д-р физ.-мат. наук, доцент
_____А.Н. Моисеев
подпись
«_23_» _мая_____ 2022 г.

Автор работы
студент группы № 931801
_____ А.А. Юдаков
подпись
«_23_» _мая_____ 2022 г.

Томск – 2022

Министерство науки и высшего образования Российской Федерации.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ООП
д-р техн. наук, профессор
_____ А.В. Замятин
подпись
«25» октября 2021 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра / специалиста /магистра обучающемуся
Юдакову Алексею Александровичу
Фамилия Имя Отчество обучающегося
по направлению подготовки Код Наименование направления подготовки, направленность (профиль) «Наименование образовательной программы»

1 Тема выпускной квалификационной работы
Разработка шлюза к ClickHouse в системе обработки результатов экспериментов A/B-тестирования

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис / деканат – 23.05.2022 б) в ГЭК – 08.06.2022

3 Исходные данные к работе:
Объект исследования – Система обработки результатов экспериментов A/B-тестирования
Предмет исследования – Разработка шлюза к ClickHouse
Цель исследования – Разработать шлюз к ClickHouse в системе обработки результатов экспериментов A/B-тестирования

Задачи:
Выполнить анализ требований, составить модель предметной области, спроектировать шлюз к ClickHouse, реализовать шлюз к ClickHouse

Методы исследования:
Анализ, синтез, классификация, абстрагирование, формализация, сравнение, эксперимент, измерение и практическое моделирование

Организация или отрасль, по тематике которой выполняется работа, –
Компания Яндекс.Технологии

4 Краткое содержание работы
Был выполнен анализ требований, составлена модель предметной области. На основе полученной модели был спроектирован и реализован шлюз к ClickHouse

Руководитель выпускной квалификационной работы
зав. каф. программной инженерии / _____ / А.Н. Моисеев
должность, место работы *подпись* *И.О. Фамилия*

Задание принял к исполнению
студент гр. 931801 / _____ / А.А. Юдаков
должность, место работы *подпись* *И.О. Фамилия*

АННОТАЦИЯ

Выпускная квалификационная работа объемом 43 страницы содержит 12 рисунков, 6 таблиц, список использованных источников из 21 наименования.

Объектом исследования является система обработки результатов экспериментов A/B-тестирования в компании Яндекс.

Предметом исследования является разработка шлюза к ClickHouse. Шлюз – это один из паттернов проектирования.

Цель исследования – разработать шлюз к ClickHouse в системе обработки результатов экспериментов A/B-тестирования.

Задачи исследования – выполнить анализ требований, составить модель предметной области, спроектировать и реализовать шлюз к ClickHouse.

Методы исследования: анализ, синтез, классификация, абстрагирование, формализация, сравнение, эксперимент, измерение и практическое моделирование.

В ходе работы был выполнен анализ требований, составлена модель предметной области. На основе полученной модели был спроектирован и реализован шлюз к ClickHouse.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Яндекс – это большая технологическая компания, широко известная в России и странах СНГ.

У компании Яндекс существует множество внешних продуктов, такие как поиск, Яндекс.Такси, Яндекс.Еда, Яндекс.Дзен и многие другие. Помимо внешних продуктов у компании есть ряд внутренних продуктов для собственного использования. В частности, у Яндекса есть собственные наработки в сфере A/B-тестирования.

Уже не первый год с помощью A/B-тестирования компания Яндекс улучшает собственные продукты. При этом сами инструменты A/B-тестирования постоянно развиваются внутри компании.

В данной работе будет описано очередное улучшение инструментов A/B-тестирования. Цель работы – разработать шлюз к ClickHouse в системе обработки результатов экспериментов A/B-тестирования.

В ходе работы будет выполнен анализ требований, составлена модель предметной области. Будет рассмотрена проблема, из-за которой потребовались изменения в системе, а также альтернативные способы решения данной проблемы.

В результате система обработки результатов экспериментов A/B-тестирования станет более гибкой в выборе источника данных. А также получит ряд функциональный возможностей, связанных с выбором в качестве источника данных ClickHouse, о чем будет описано в данной работе.

1 Предметная область

1.1 Основы A/B-тестирования

Когда мы развиваем свой сервис наши предположения и гипотезы строятся на основе личного опыта и наших взглядов, которые совсем не обязательно совпадают со взглядами пользователей. То есть наша идея, которая нам кажется очень хорошей, может быть воспринята пользователями иначе. Поэтому для проверки таких идей мы и проводим A/B-тестирования.

A/B-тестирование – это инструмент позволяющий улучшать свой сервис с помощью математической статистики на основе действий пользователей. Идея A/B-тестирования довольно проста – разобьем пользователей нашего сервиса на выборки А и Б. Выборку А назовем контрольной. Для пользователей, попавших в эту выборку, мы оставим наш сервис без изменений. Выборку Б назовем экспериментальной. Для пользователей, попавших в эту выборку, мы будем включать нашу экспериментальную функциональность.

Также A/B-тестирование позволяет снизить влияние внешних факторов, таких как сезонность, рекламные кампании, день недели, праздничные дни или погода. Данное снижение достигается за счет того, что сбор данных о выборках А и Б производится параллельно, в этот момент пользователи из выборок А и Б находятся в одинаковых внешних условиях.

При проведении A/B-тестирования важно и количество пользователей сервиса. Чем больше данных удастся собрать, тем более точные будут результаты. С другой стороны, некоторые изменения в сервисе могут вызвать сильную негативную реакцию у пользователей, поэтому для таких случаев лучшее решение – показать экспериментальную функциональность на небольшой процент пользователей. В вопросе об объеме собираемых данных у нас есть выбор – увеличивать процент пользователей, на котором мы проводим наш эксперимент или увеличить время проведения эксперимента.

Обычно у сервиса есть ряд показателей, которые хочется улучшить. Если это интернет-магазин, то ключевыми показателями могут быть, например, число посетителей за день, средний чек и объем выручки. Показатели, которые имеют численное представление, называются метриками.

Как правило, до проведения A/B-тестирования есть интуитивное предположение о том, как должны измениться метрики сервиса. Во время проведения эксперимента, если разница между метриками, полученным из выборок А и Б, двигается в нужном направлении, можно захотеть закончить A/B-тестирование досрочно, сказав, что мы получили ожидаемый результат. Но здесь нужно быть осторожным, так как каждая метрика изменяется день ото дня, то есть метрика является случайно величиной. Важный вопрос, на который должен ответить человек, проводящий A/B-тестирование, это – а на сколько я уверен, что полученный результата не случаен?

Для того, чтобы быть уверенным в результате A/B-тестирования, используется аппарат математической статистики. Для каждой метрики выдвигается гипотеза, что ее распределение в выборке А и в выборке Б одинаковое. Далее определяется уровень значимости и осуществляется проверка статистических гипотез.

Для проверки статистических гипотез используются статистические тесты. Тесты выбираются в зависимости от природы данных и их объема.

В итоге, после A/B-тестирования, мы получаем метрики из выборок А и Б, разницу между ними и долю уверенности в том, что эта разница не случайна. Полученные результаты анализируются, на их основе делается вывод об экспериментальной функциональности.

1.2 A/B-тестирование в Яндексе

Многие крупные компании имеют собственные инструменты в сфере A/B-тестирования. Некоторые компании предоставляют эти инструменты

внешним пользователем, например, Google предоставляет Google Analytics Experiments [1].

Яндекс также имеет собственные инструменты для проведения A/B-тестирования. Для этого есть отдельная команда, занимающаяся разработкой инфраструктуры экспериментов. В Яндексе эксперименты проводятся централизовано. Такой подход имеет ряд преимуществ:

- Использование лучших практик в A/B-тестировании. Со временем приходит понимание, как лучше проводить A/B-тестирование, поэтому новые сервисы, которые до этого ни разу не проводили A/B-тестирование, могут сразу проводить свои эксперименты наиболее эффективно, с использованием лучших практик.
- Переиспользование кода. Если метрики для каждого сервиса могут быть уникальными, то способ сравнения выборок между друг другом, проверка статистических гипотез, интерфейс для просмотра метрик и прочее может быть унифицированным и единым для всех сервисов.
- Снижение порога входа. Яндекс – большая компания, содержащая в себе множество сервисов различной величины. Для крупных сервисов выделить группу разработчиков для проведения экспериментов не проблема. Но для мелких сервисов, где на счету каждый сотрудник, гораздо проще воспользоваться готовым решением, нежели придумывать свое решение.
- Межсервисное A/B-тестирование. Иногда экспериментальная функциональность влияет не только на собственный сервис, но и на смежные сервисы. Например, если поиск Яндекса перестанет выдавать ссылки на Яндекс.Маркет, то такое экспериментальное изменение может существенно изменить метрики Яндекс.Маркета. При централизованном подходе проводить межсервисные эксперименты проще.

Рассмотрим схему A/B-тестирования в Яндексе более подробно. Условно эту схему можно разбить на два этапа: сбор данных и анализ данных. Схему сбора данных можно увидеть на рисунке 1, а схему анализа данных – на рисунке 2.

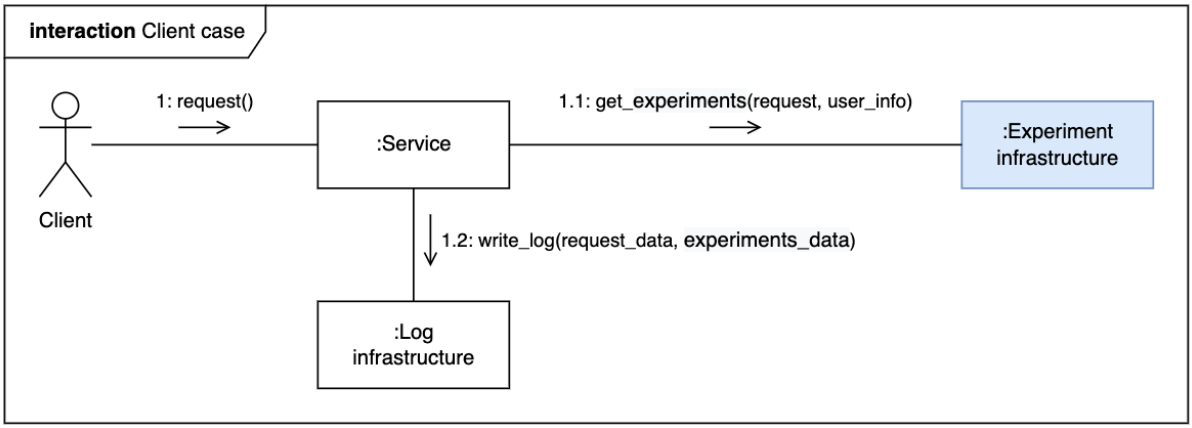


Рисунок 1 – Диаграмма сбора данных

Рассмотрим процесс сбора данных (Рисунок 1) по шагам:

1 Клиент нашего сервиса отправил некоторый запрос.

2 Чтобы ответить на запрос клиента нужно понять, в какие эксперименты он попал. Для этого сервис запрашивает данную информацию у инфраструктуры экспериментов.

3 На основе информации о том, в какие A/B-тестирования попал пользователь, сервис можем корректировать свой ответ, включая нужную экспериментальную функциональность.

4 Затем сервис должен залогировать действия пользователя и информацию об экспериментах, в которые попал пользователь.

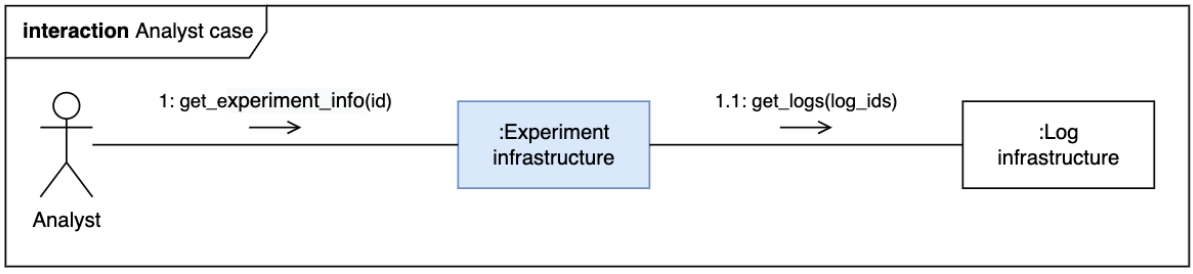


Рисунок 2 – Диаграмма анализа данных

Теперь рассмотрим процесс анализа данных (Рисунок 2):

1 После или во время проведения A/B-тестирования аналитик сервиса, в котором проводится эксперимент, получает из инфраструктуры экспериментов результаты A/B-тестирования.

2 В свою очередь инфраструктура экспериментов для того, чтобы предоставить результаты должна обработать логи сервиса. То есть получить из логов нужные аналитику метрики. После этого аналитик может приступить к анализу результатов эксперимента.

2 Анализ требований

2.1 Функциональные требования

Основные пользователи инфраструктуры экспериментов — это аналитики. Рассмотрим их требования, которые возникают в процессе анализа результатов эксперимента.

Первое требование – это возможность задавать способ преобразования логов в свойства. Свойства являются базовыми единицами, на основе которых строятся метрики. Логи представляют из себя схематизированные таблицы.

Второе требование – возможность вести список метрик. На этапе преобразования логов мы хотим получить таблицу с двумя столбцами, первый столбец – название свойства, второй столбец – значение свойства. Далее аналитик должен иметь возможность на основе свойств задавать метрики. В общем случае метрика является произвольной формулой над свойствами. Например, если мы возьмем за свойства «количество пользователей» и «количество запросов к сервису», то в качестве метрики мы можем получить «среднее количество запросов на пользователя», разделив одно свойство на другое. Промежуточный слой в виде свойств позволяет, не изменяя кода расчета, создавать новые метрики, а также экономить память, так как количество свойств всегда меньше или равно количеству метрик.

Третье требование – возможность смотреть результаты A/B-тестирования. Под этим требованием подразумевается сразу несколько вещей. Аналитик должен получать полный отчет по эксперименту, в том числе: значения метрик в выборках А и Б, абсолютная разница между метриками, разница между метриками в процентном соотношении, результаты статистических критериев для заданного уровня значимости. А также у аналитика должна быть возможность посмотреть метрики не только по всем пользователям, попавшим в эксперимент, но и по определенной части (срезу) пользователей.

2.2 Нефункциональные требования

При выборе способа преобразовании логов в свойства необходимо рассматривать популярные среди аналитиков инструменты. Например, преобразование может задаваться кодом на языке программирования Python или с помощью запросов к СУБД на языке SQL.

3 Анализ текущего решения

3.1 Система обработки результатов экспериментов

Для удовлетворения функциональных требований уже существует система обработки результатов экспериментов. Рассмотрим окружение этой системы на рисунке 3.

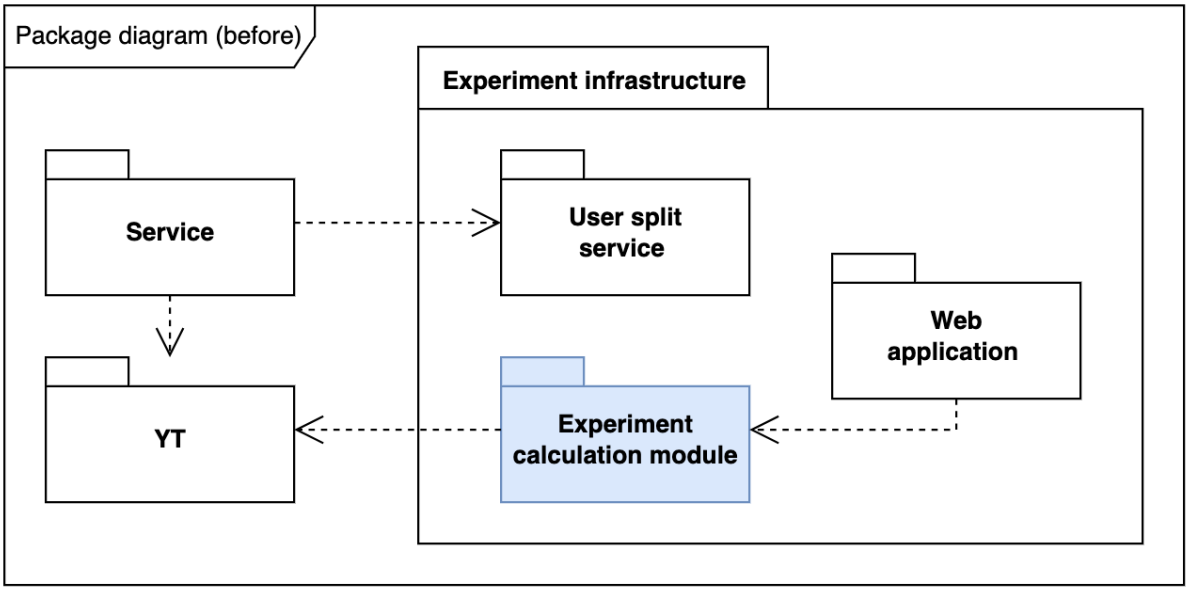


Рисунок 3 – Окружение системы обработки результатов экспериментов

Диаграммы на рисунках 1, 2 и 3 описывают один процесс. Разница в том, что диаграммы на рисунках 1 и 2 являются поведенческими, а диаграмма на рисунке 3 является структурной. Рассмотрим подробнее пакеты, изображенные на рисунке 3:

- Service – некоторый сервис, который хочет проводить A/B-тестирование. Например, Яндекс.Еда или Яндекс.Маркет.
- YT – внутренняя инструмент Яндекса, позволяющий хранить структурированные данные, а также запускать над этими данными MapReduce-операции.
- User split service – сервис из инфраструктуры экспериментов, который разделяет пользователей на эксперименты, в которые они попадают.

- Web application – единое для всех экспериментаторов веб-приложение. В нем аналитик может посмотреть информацию о своих A/B-тестированиях, увидеть их результаты.
- Experiment calculation module – это система обработки результатов экспериментов. Именно в этом месте предстоит произвести изменения.

Теперь рассмотрим связи между пакетами, изображенными на рисунке 3. Service взаимодействует с User split service и YT. Когда в сервис приходит запрос от пользователя, чтобы узнать, в какие эксперименты попал пользователь, сервис использует User split service. Далее, чтобы залогировать действия пользователя сервис использует YT.

Когда аналитик хочет посмотреть результаты A/B-тестирования, он должен воспользоваться Web application. В свою очередь Web application использует Experiment calculation module, который на основе логов, хранящихся в YT, рассчитывает метрики, запрошенные аналитиком.

3.2 Возможности и недостатки YT

Достаточно важным звеном в системе обработки результатов экспериментов является YT. Во-первых, в этой системе хранятся исходные логи. Во-вторых, в системе YT происходит их обработка. Сама обработка логов основана на парадигме MapReduce.

MapReduce – это модель распределенных вычислений, представленная компанией Google, используемая для параллельных вычислений над очень большими, вплоть до нескольких петабайт, наборами данных в компьютерных кластерах [2].

Для того чтобы воспользоваться YT достаточно написать собственные MapReduce-операции на C++ или Python. С одной стороны, YT позволяет перерабатывать петабайты данных. Но, с другой стороны, для аналитиков писать MapReduce-операции может оказаться нетривиальной задачей.

Также в силу того, что YT основан на MapReduce, он не может использоваться в качестве бэкенда для веб-интерфейса. То есть из-за высоких задержек между отправкой запроса на расчет в YT и получением результата мы вынуждены заранее предрассчитывать результаты всех экспериментов.

Здесь стоит отметить, что аналитики смотрят не только на метрики по всем пользователем, но и по определенным срезам. Получается так, что из-за предрасчета данных мы должны заранее знать список этих срезов. Если во время анализа результатов эксперимента потребуется новый срез, который не был заведен заранее, то на пересчет данных для этого среза потребуется продолжительно время.

4 Анализ альтернативных решений

4.1 Выбор альтернативного решения

Для начала сформулируем главное требование к альтернативному решению – это возможность отвечать на запросы с небольшой задержкой, секунды или единицы минут. Рассматривать Apache Hadoop [3], Apache Spark [4] или любую другую систему основанную на MapReduce смысла нет, так как сама технология MapReduce не заточена под ответы на запросы с минимальной задержкой.

Тогда можно посмотреть в сторону СУБД. Так как логи действий пользователя обычно хорошо структурированы, то стоит рассмотреть СУБД, которые хранят данные в табличном виде.

Так как анализ альтернативных решений проводится в контексте компании Яндекс, то имеет смысл сразу сузить выбор СУБД, обратившись к Yandex Cloud. Список сервисов, которые предоставляет Yandex Cloud, представлены на рисунке 4.

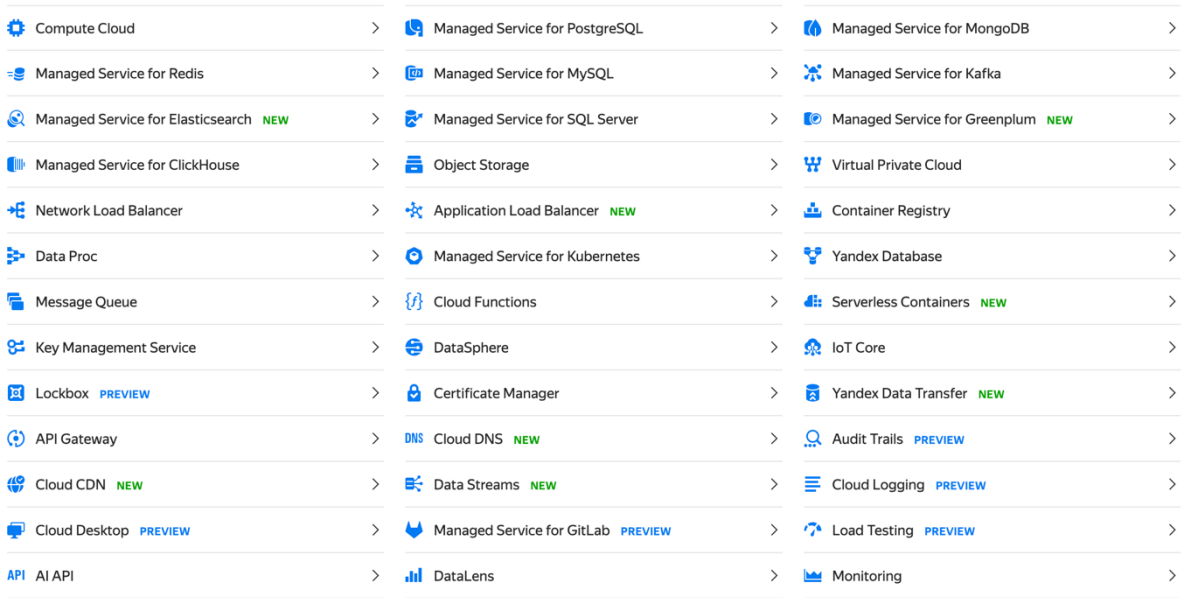


Рисунок 4 – Сервисы, предоставляемые Yandex Cloud [5]

Выберем из этого списка СУБД и оценим, имеет ли смысл рассматривать дальше эту СУБД для хранения и обработки логов.

- Redis (от англ. remote dictionary server) — резидентная система управления базами данных класса NoSQL с открытым исходным кодом, работающая со структурами данных типа «ключ — значение» [6]. Используется как для баз данных, так и для реализации кэшей, брокеров сообщений. Не подходит для хранения и обработки логов.
- ClickHouse — это колоночная аналитическая СУБД с открытым кодом, позволяющая выполнять аналитические запросы в режиме реального времени на структурированных больших данных, разрабатываемая компанией Яндекс [7]. Хороший кандидат для наших задач.
- PostgreSQL — свободная объектно-реляционная система управления базами данных [8]. Содержит высокопроизводительные и надежные механизмы транзакций и репликации. Больше подходит для хранения данных бизнес-логики, чем для хранения логов.
- MySQL — свободная реляционная система управления базами данных [9]. Разработку и поддержку MySQL осуществляет корпорация Oracle. MySQL является решением для хранения бизнес-логики малых и средних приложений. Не очень подходит под наши задачи.
- MongoDB — документоориентированная система управления базами данных, не требующая описания схемы таблиц [10]. Является NoSQL-системой, использует JSON-подобные документы и схему базы данных. Не подходит для наших задач.
- Greenplum — это СУБД для обработки больших данных, основанная на архитектуре MPP и технологии баз данных с открытым исходным кодом Postgres [11]. Может подойти для задач хранения и обработки логов.

4.2 Сравнение СУБД

Выберем для сравнения все четыре SQL базы данных из представленных выше – ClickHouse, PostgreSQL, MySQL и Greenplum. Сравним эти базы данных по возможностям, которые предоставляет Yandex Cloud по развертыванию этих СУБД, их цену за месяц использования и производительность на SQL-запросах, которые могут возникать при расчете результатов экспериментов.

Возможности, которые предоставляет Yandex Cloud по развертыванию сравниваемых СУБД представлены в таблице 1. Как видно из таблицы, ClickHouse, PostgreSQL и MySQL могут быть развернуты в Yandex Cloud более гибко, как с точки зрения выбора хранилища данных, так и с точки зрения конфигурации хоста. Greenplum появился в Yandex Cloud сравнительно недавно, возможно именно по этой причине он уступает остальным СУБД.

Таблица 1 – Возможности по развертыванию СУБД в Yandex Cloud

	СУБД			
	ClickHouse	PostgreSQL	MySQL	Greenplum
Можно выбрать Network-ssd хранище	Да	Да	Да	Нет
Можно выбрать Network-hdd хранище	Да	Да	Да	Нет
Можно выбрать Local-ssd хранилище	Да	Да	Да	Да
Количество конфигураций хоста	71	70	70	10

Теперь рассмотрим цену за месяц использования СУБД. Так как для развертывания Greenplum можно выбрать только local-ssd хранилище, то возьмем именно его, в размере 200 ГБ. Данные о стоимости использования СУБД представлены в таблице 2. Из таблицы видно, что при прочих равных условиях стоимость развертывания ClickHouse самая дешевая, а Greenplum

сама дорогая. Стоимость развертывания PostgreSQL и MySQL примерно равна, но MySQL чуть дороже.

Таблица 2 – Стоимость развертывания СУБД в Yandex Cloud

Конфигурация хоста	Стоимость, руб/мес			
	ClickHouse	PostgreSQL	MySQL	Greenplum
2 ядра, 8 ГБ RAM	23 248.70	24 092.40	24 697.20	-
8 ядер, 32 ГБ RAM	57 808.70	72 951.60	75 370.80	120 932.40
12 ядер, 48 ГБ RAM	80 848.70	105 524.40	109 153.20	149 214.00
32 ядра, 128 ГБ RAM	196 048.70	268 388.40	278 065.20	-
64 ядра, 256 ГБ RAM	380 368.70	528 970.80	548 324.40	-

В заключении сравним СУБД по производительности. Из-за ограничений Yandex Cloud на использование большого количества ресурсов без дополнительной верификации не получится развернуть все СУБД одновременно. Поэтому было принято решение сравнить ClickHouse, PostgreSQL и MySQL в конфигурации хоста 2 ядра, 8 ГБ RAM, а затем лучшую базу с Greenplum в конфигурации хоста 8 ядер, 32 ГБ RAM.

Для проведения тестирования в Yandex Cloud была создана виртуальная машина, из которой происходили все замеры. Также была написана программа на Python, которая умеет замерять скорость выполнения SQL-запросов. В качестве тест-кейсов использовались SELECT запросы с WHERE, HAVING, ORDER BY и агрегатными функциями. Такие запросы больше всего похожи на то, что делают аналитики в реальной жизни. Также, помимо SELECT запросов, было замерено время выполнения INSERT INTO запросов, которые вызывались при загрузке данных в СУБД.

В тестах данные генерировались случайном образом, далее они загружались во все СУБД с замером скорости загрузки. Таким образом в момент выполнения SELECT запросов данные в СУБД были одинаковые. Также, в момент проведения тестов, все СУБД были развернуты на хостах с одинаковыми характеристиками.

Ниже приведена схема данных тестовой таблицы. В силу того, что каждая СУБД имеет собственные названия для типов данных, в описании ниже были использованы типы данных из PostgreSQL [12], для остальных СУБД были использованы максимально похожие типы.

- Timestamp – integer поле, содержит число от 1650855600 до 1650891600
- UserId – char(16) поле, содержит строку длины 16
- RequestId – char(32) поле, содержит строку длины 32
- ClickType – smallint поле, содержит число от 1 до 5
- ClickCost – integer поле, содержит число от 0 до 1000
- ElementType – smallint поле, содержит число от 1 до 50
- ElementName – varchar(200) поле, содержит строку длины от 15 до 100
- Position – smallint поле, содержит число от 0 до 100

Перейдем к сравнению ClickHouse, PostgreSQL и MySQL. Для этого развернем все три базы на хосте с конфигурацией 2 ядра, 8 ГБ RAM и 10 ГБ network-ssd. Данные были загружены по 5 000 записей за один запрос, всего было вызвано 2 000 запросов INSERT INTO. Итоговый объем данных составил 10 000 000 записей. Результаты тестирования представлены в таблице 3.

Таблица 3 – Тест производительности ClickHouse, PostgreSQL и MySQL

Запрос	Время, секунд		
	ClickHouse	PostgreSQL	MySQL
INSERT INTO t …	158.631	331.075	389.925
SELECT * FROM t ORDER BY ClickCost LIMIT 5	0.916	1.508	9.122
SELECT count(*) FROM t WHERE ClickCost > 10	0.070	1.283	5.845
SELECT min(Timestamp), max(Timestamp) FROM t	0.049	1.325	6.114
SELECT * FROM t WHERE ClickType = 3 ORDER BY ClickCost LIMIT 5	1.069	1.163	8.349

Продолжение таблицы 3

Запрос	Время, секунд		
	ClickHouse	PostgreSQL	ClickHouse
SELECT UserID, sum(ClickCost), count(ClickCost) FROM t WHERE ClickCost > 0 GROUP BY UserID LIMIT 5	4.372	40.614	207.534
SELECT UserID, sum(ClickCost) FROM t WHERE ClickType = 2 GROUP BY UserID ORDER BY sum(ClickCost) LIMIT 5	0.898	10.009	11.054
SELECT UserID, sum(ClickCost), count(ClickCost) FROM t GROUP BY UserID HAVING sum(case when ClickType = 1 then 1 end) > 0 ORDER BY sum(ClickCost) LIMIT 5	4.423	46.315	415.219
Среднее время на SELECT	1.685	14.602	94.748
Нормализованное время на INSERT INTO	1.000	2.087	2.458
Нормализованное среднее время на SELECT	1.000	8.665	56.230

Обратим внимание на последние две строки из таблицы 3. Получается, что выполнение вставки 5 000 записей ClickHouse осуществляет в 2 раза быстрее PostgreSQL и в 2.5 раза быстрее MySQL. В среднем времени на запрос также выигрывает ClickHouse, он выполняет запрос в среднем в 8.7 раз быстрее PostgreSQL и в 56.2 раза быстрее MySQL.

В предыдущем сравнении лучше всех себя проявил ClickHouse, поэтому теперь сравним производительность ClickHouse и Greenplum. Развернем обе СУБД на хосте с конфигурацией 8 ядер, 32 ГБ RAM и 100 ГБ local-ssd. Данные будем загружать по 10 000 записей за раз, всего в таблице будет 100 000 000 записей. Результаты тестирования представлены в таблице 4.

Таблица 4 – Тест производительности ClickHouse и Greenplum

Запрос	Время, секунд	
	ClickHouse	Greenplum
INSERT INTO t …	92.490	965.112
SELECT * FROM t ORDER BY ClickCost LIMIT 5	0.263	1.665
SELECT count(*) FROM t WHERE ClickCost > 10	0.040	0.932
SELECT min(Timestamp), max(Timestamp) FROM t	0.033	0.862
SELECT * FROM t WHERE ClickType = 3 ORDER BY ClickCost LIMIT 5	0.317	0.871
SELECT UserID, sum(ClickCost), count(ClickCost) FROM t WHERE ClickCost > 0 GROUP BY UserID LIMIT 5	0.940	5.502
SELECT UserID, sum(ClickCost) FROM t WHERE ClickType = 2 GROUP BY UserID ORDER BY sum(ClickCost) LIMIT 5	0.200	1.786
SELECT UserID, sum(ClickCost), count(ClickCost) FROM t GROUP BY UserID HAVING sum(case when ClickType = 1 then 1 end) > 0 ORDER BY sum(ClickCost) LIMIT 5	1.079	7.607
Среднее время на SELECT	0.410	2.746
Нормализованное время на INSERT INTO	1.000	10.434
Нормализованное среднее время на SELECT	1.000	6.693

Обратим внимание на последние две строки из таблицы 4. Получается, что выполнение вставки 10 000 записей ClickHouse осуществляет в 10.4 раза быстрее Greenplum. В среднем времени на запрос также выигрывает ClickHouse, он выполняет запрос в среднем в 6.7 раз быстрее Greenplum.

Также на официальном сайте ClickHouse есть тесты со сравнением производительности (Рисунок 5). Их тесты также показывают, что ClickHouse выигрывает в производительности у Greenplum, PostgreSQL и MySQL, причем с большим отрывом, если сравнивать с тестами, проведенными в этой работе. Но здесь стоит отметить, что у них была другая схема данных, были другие запросы и было другое оборудование, на котором были развернуты СУБД.

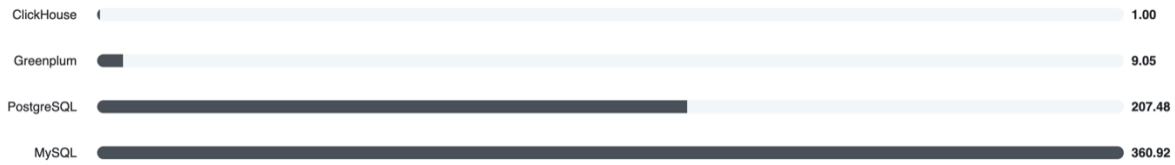


Рисунок 5 – Относительное время обработки запроса (чем меньше, тем лучше) [13]

В заключении можно сказать, что ClickHouse выигрывает своих конкурентов по всем параметрам. Он выгоднее в экономическом плане и быстрее работает. Поэтому ClickHouse был выбран в качестве альтернативного решения.

4.3 Сравнение YT и ClickHouse

Сравним возможности YT и ClickHouse в задачах системы обработки результатов экспериментов A/B-тестирования.

Начнем со способа обработки данных. В YT необходимо писать MapReduce-операции на C++ или Python. В ClickHouse для обработки данных пишутся SQL-запросы. С точки зрения эффективности обработки операций выигрывает ClickHouse, так как, во-первых, аналитики пишут MapReduce-операции на Python, который сильно медленнее C++, а во-вторых, SQL-запросы можно оптимизировать. Стоит добавить, что для большинства аналитиков написание SQL-запросов более предпочтительный вариант обработки логов, нежели написание Python кода.

Теперь рассмотрим объем обрабатываемых данных. Разработчики YT заявляют, что их система может обрабатывать единицы петабайт данных. Для ClickHouse эта цифра порядка единиц терабайтов. В действительности большинство таблиц с логами имеют размер порядка десятков гигабайт. Хотя есть и исключения в виде крупных сервисов, у которых логи занимают порядка единиц терабайт, например, один из дневных логов рекламы занимает в среднем около 11 терабайт места на диске. Для таких крупных логов

использование YT становится безальтернативным, для логов поменьше можно использовать ClickHouse.

Одно из преимуществ ClickHouse над YT – это способность быстро отвечать на запросы. Средний запрос к YT занимает десятки минут, отдельные, крупные расчеты, могут занимать часы. ClickHouse отвечает на запросы за единицы секунд. Это свойство позволяет производить расчеты по требованию, а не предварительно, как в случае с YT из-за большой задержки.

Следующее отличие заключается в свойствах, которые можно посчитать с помощью YT или ClickHouse. В случае YT мы пишем цепочку MapReduce-операций на Python, что позволяет считать свойства произвольной сложности. Например, с помощью YT мы можем объединять запросы пользователя в сессии, для этого скажем, что запросы попадают в одну сессию, если разница между ними не более 5 минут. Не сложно написать код на Python, который решает задачу объединения запросов в пользовательские сессии, а затем считает некоторые сессионные свойства. В ClickHouse мы ограничены языком SQL, во-первых, некоторые запросы, например выделение сессий, могут быть настолько сложными, что только автор сможет разобраться в них, во-вторых, ClickHouse выполняет запросы в памяти, поэтому он не сможет выполнить слишком сложные запросы.

В итоге ни одна из систем не может заменить другую. Если у сервиса небольшой объем логов и он не хочет считать сложные метрики, то этому сервису лучше использовать ClickHouse. В другом случае, если у сервиса большие объемы логов, или он хочет считать сложные метрики, то этому сервису нужно использовать YT. По этой причине в системе расчетов результатов экспериментов A/B-тестирования необходимо поддерживать оба источника данных.

5 Технологический стек

5.1 Python

В качестве языка программирования для реализации системы обработки результатов экспериментов A/B-тестирования был выбран Python.

Python – это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нем программ [14].

Python является полностью объектно-ориентированным языком в том плане, что все является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счет чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов.

Недостатками языка Python являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нем программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++ [15].

Основная причина, по которой был выбран Python – это необходимость дать аналитику возможность задавать способ преобразования логов в свойства. Если бы аналитику пришлось писать код, скажем, на C++, то это бы значительно повысило порог вхождения в систему обработки результатов экспериментов A/B-тестирования.

5.1.1 Python-библиотека requests

Requests - это библиотека для языка программирования Python, позволяющая просто и удобно использовать HTTP-запросы [16]. Данная

библиотека выпущена под лицензией Apache 2.0, которая дает пользователю право использовать эту библиотеку для любых целей.

Системе обработки результатов экспериментов A/B-тестирования необходима возможность взаимодействовать с ClickHouse. Yandex Cloud предлагает подключиться к СУБД с помощью HTTP протокола. ClickHouse не предоставляет официальную реализацию клиента к СУБД на языке Python. Поэтому библиотека requests будет использоваться при написании клиента к ClickHouse.

5.1.2 Python-библиотека json

Json - это встроенная в язык программирования Python библиотека, позволяющая удобно работать с текстовым форматом обмена данными с одноименным названием json [17]. Данная библиотека предоставляет два основных метода. Метод dump сериализует словарь или список из Python в json строку. Метод load десериализует json строку в Python объект.

5.2 ClickHouse

ClickHouse - столбцовая система управления базами данных для онлайн обработки аналитических запросов (OLAP) [18]. Разница между строковой (Таблица 5) и столбцовой (Таблица 6) СУБД заключается в способе хранения данных. В строковых СУБД значения, относящиеся к одной строке, физически хранятся рядом. В столбцовых СУБД значения, относящиеся к одной строке, хранятся отдельно, а данные одного столбца - вместе.

Таблица 5 – Пример хранения данных в строковой СУБД

Строка	Столбец				
	WatchID	JavaEnable	Title	GoodEvent	EventTime
#0	89354350662	1	Investor Relations	1	2016-05-18 05:19:20
#1	90329509958	0	Contact us	1	2016-05-18 08:10:20
#2	89953706054	1	Mission	1	2016-05-18 07:38:00
#N	…	…	…	…	…

Таблица 6 – Пример хранения данных в столбцовой СУБД

Столбец	Строка			
	#0	#1	#2	#N
WatchID	89354350662	90329509958	89953706054	…
JavaEnable	1	0	1	…
Title	Investor Relations	Contact us	Mission	…
GoodEvent	1	1	1	…
EventTime	2016-05-18 05:19:20	2016-05-18 08:10:20	2016-05-18 07:38:00	…

Разный порядок хранения данных лучше подходит для разных сценариев работы. Сценарий работы с данными - это то:

- какие производятся запросы, как часто и в каком соотношении;
- сколько читается данных на запросы каждого вида - строк, столбцов, байт;
- как соотносятся чтения и обновления данных;
- какой рабочий размер данных и насколько локально он используется; используются ли транзакции и с какой изолированностью;
- какие требования к дублированию данных и логической целостности; требования к задержкам на выполнение и пропускной способности запросов каждого вида;
- различные другие требования.

5.2.1 OLAP-сценарии работы

Чем больше нагрузка на систему, тем более важной становится специализация под сценарий работы, и тем более конкретной становится эта специализация. Не существует системы, одинаково хорошо подходящей под существенно различные сценарии работы. Если система подходит под широкое множество сценариев работы, то при достаточно большой нагрузке, система будет справляться со всеми сценариями работы плохо, или справляться хорошо только с одним из сценариев работы.

ClickHouse отлично справляется с OLAP-сценариями работы, ниже перечислены их ключевые особенности:

- подавляющее большинство запросов - на чтение;
- данные обновляются достаточно большими пачками (> 1000 строк), а не по одной строке, или не обновляются вообще;
- данные добавляются в БД, но не изменяются;
- при чтении вынимается достаточно большое количество строк из БД, но только небольшое подмножество столбцов;
- таблицы являются «широкими», то есть, содержат большое количество столбцов;
- запросы идут сравнительно редко (обычно не более сотни в секунду на сервер);
- при выполнении простых запросов, допустимы задержки в районе 50 мс;
- значения в столбцах достаточно мелкие - числа и небольшие строки (пример - 60 байт на URL);
- требуется высокая пропускная способность при обработке одного запроса (до миллиардов строк в секунду на один сервер);
- транзакции отсутствуют;
- низкие требования к консистентности данных;
- в запросе одна большая таблица, все таблицы кроме одной маленькие;

- результат выполнения запроса существенно меньше исходных данных - то есть, данные фильтруются или агрегируются; результат выполнения помещается в оперативную память на одном сервере.

Можно сказать, что система обработки результатов экспериментов A/B-тестирования выполняет OLAP-сценарий работы. Логи никогда не изменяются, транзакции не нужны, часть логов можно утратить, лог – это одна большая таблица и так далее. Именно поэтому ClickHouse так хорошо справился с тестами по сравнению с конкурентами.

5.2.1 Формат выходных данных в ClickHouse

ClickHouse имеет множество форматов для входных и выходных данных. Так как система обработки результатов экспериментов A/B-тестирования будет только читать данные с ClickHouse, то нас интересуют форматы выходных данных.

Всего ClickHouse предоставляет 52 формата выходных данных [19], из них, например, самые популярные CSV, JSON, TSKV, Protobuf и XML. В ходе анализа было принято решение использовать формат JSONCompact (Рисунок 6).

```
{
        "meta":
        [
                {
                        "name": "'hello'",
                        "type": "String"
                },
                {
                        "name": "multiply(42, number)",
                        "type": "UInt64"
                },
                {
                        "name": "range(5)",
                        "type": "Array(UInt8)"
                }
        ],

        "data":
        [
                ["hello", "0", [0,1,2,3,4]],
                ["hello", "42", [0,1,2,3,4]],
                ["hello", "84", [0,1,2,3,4]]
        ],

        "rows": 3,

        "rows_before_limit_at_least": 3
}
```


Рисунок 6 - Пример данных в формате JSONCompact

Выбранный формат похож на JSON (Рисунок 7), но в отличии от JSON выбранный JSONCompact более экономичный. Также json-формат легко читается человеком, а значит его можно без особых проблем составлять самому, например, для тестов. Также с данным форматом легко работать с помощью встроенной в Python одноименной библиотека json.

```
{
        "meta":
        [
                {
                        "name": "'hello'",
                        "type": "String"
                },
                {
                        "name": "multiply(42, number)",
                        "type": "UInt64"
                },
                {
                        "name": "range(5)",
                        "type": "Array(UInt8)"
                }
        ],

        "data":
        [
                {
                        "'hello'": "hello",
                        "multiply(42, number)": "0",
                        "range(5)": [0,1,2,3,4]
                },
                {
                        "'hello'": "hello",
                        "multiply(42, number)": "42",
                        "range(5)": [0,1,2,3,4]
                },
                {
                        "'hello'": "hello",
                        "multiply(42, number)": "84",
                        "range(5)": [0,1,2,3,4]
                }
        ],

        "rows": 3,

        "rows_before_limit_at_least": 3
}
```


Рисунок 7 - Пример данных в формате JSON

6 Проектирование

6.1 ClickHouse в окружении системы обработки результатов экспериментов

На рисунке 3 было представлено окружение системы обработки результатов экспериментов. Добавим в это окружение ClickHouse. Полученная диаграмма представлена на рисунке 8.

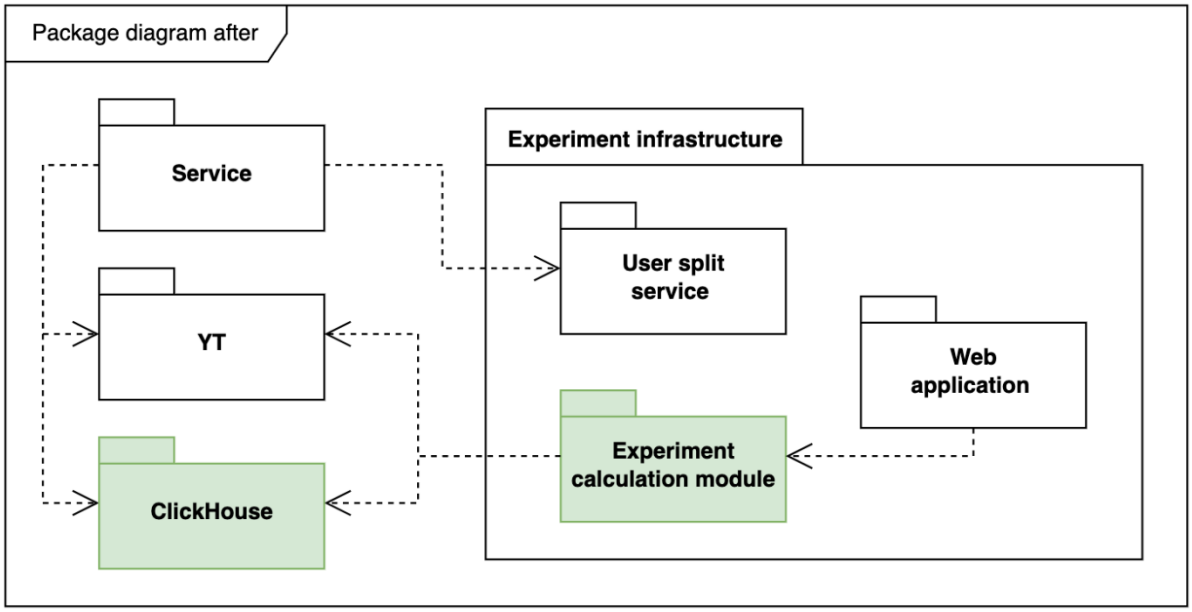


Рисунок 8 – Окружение системы обработки результатов экспериментов с ClickHouse

После добавления ClickHouse в окружение системы обработки результатов экспериментов сервис, который до этого мог записывать логи только на YT, теперь может записывать логи в ClickHouse. В свою очередь система обработки результатов экспериментов может обрабатывать логи не только из YT, но и из ClickHouse. Все остальные части окружения остались без изменений.

6.2 Система обработки результатов экспериментов до изменений

В функциональных требованиях сказано, что аналитик должен иметь возможность задавать способ преобразования логов в свойства. Для этого в системе обработки результатов экспериментов есть базовые классы, от которых может наследоваться аналитик. В наследованных классах можно указать собственную логику обработки входных логов. Классы, доступные для наследования, представлены на рисунке 9.

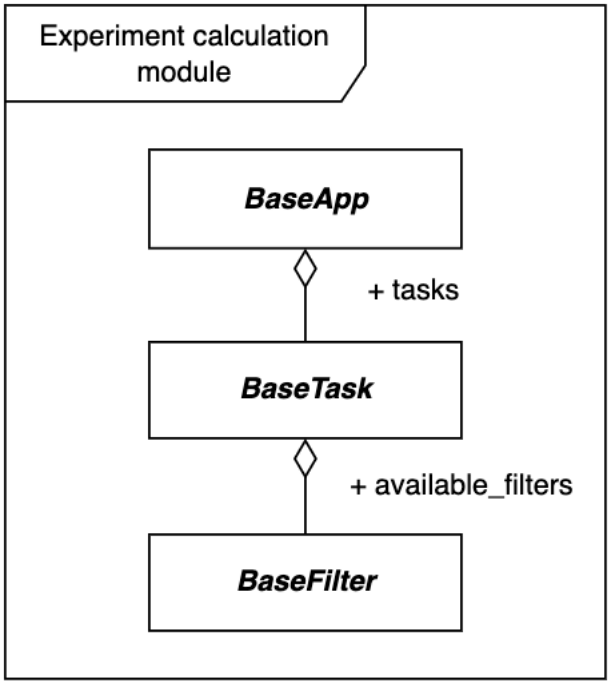


Рисунок 9 – Классы, доступные для наследования

Разберем рисунок 9 подробнее. BaseApp – это базовый класс проекта, нужен для логического объединения группы расчетов в один проект. BaseTask – это базовый класс расчета, в данном классе содержится вся логика преобразования входных логов в свойства. BaseFilter – это базовый класс фильтра. Фильтры нужны для создания срезов, это одно из требований аналитиков к системе обработки результатов экспериментов.

6.3 Система обработки результатов экспериментов после изменений

Добавим в систему обработки результатов экспериментов возможность взаимодействовать с ClickHouse. Рассмотрим, какие теперь классы доступны для наследования аналитику. Полученные изменения представлены на рисунке 10.

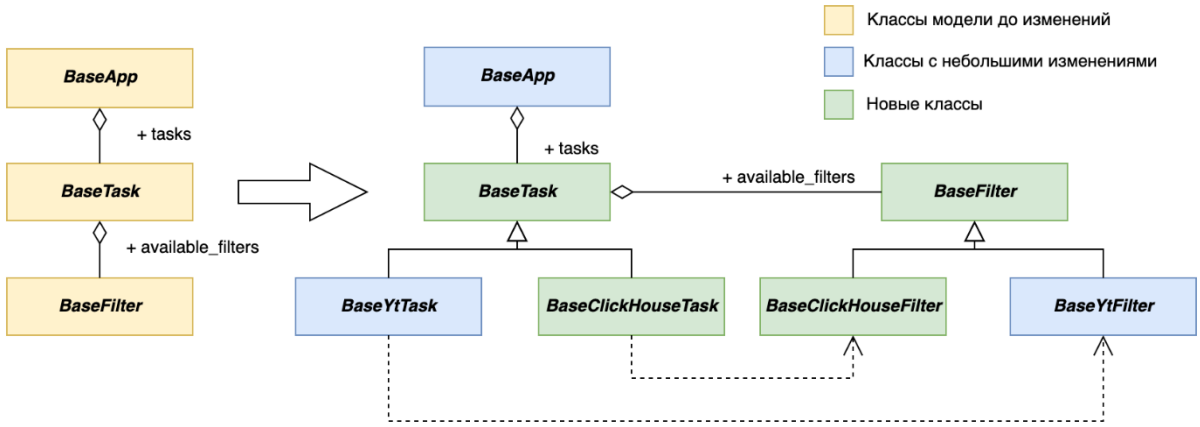


Рисунок 10 – Изменения в классах, доступных для наследования

Рассмотрим изменения подробней. Класс BaseApp остался почти без изменений, этот класс объединяет различные расчеты проекта в одну сущность. До изменений классы BaseTask и BaseFilter были нужны для взаимодействия с YT, поэтому после изменений эти классы были переименованы в BaseYtYask и BaseYtFilter. Также появились новые классы – BaseClickHouseTask и BaseClickHouseFilter. Эти классы нужны для взаимодействия с ClickHouse. Вся общая логика между Yt и ClickHouse классами была вынесена в абстрактные классы – BaseTask и BaseFilter.

Иными словами, у аналитика появилась возможность обрабатывать логи не только в YT, но и в ClickHouse, для этого аналитику достаточно унаследоваться от нужного класса.

6.4 Паттерн шлюз

Паттерн шлюз был предложен Мартином Фаулером [20]. Основная идея паттерна заключается в том, чтобы предоставить класс, который инкапсулирует в себе взаимодействие с внешней системой.

Современное ПО редко функционирует в изоляции от внешнего мира. Даже самая строго объектно-ориентированная система часто вынуждена взаимодействовать с "не объектами", например реляционная БД, CICS транзакции или структурами XML.

При доступе к такого рода внешним ресурсам, обычно используется API. Однако, API изначально являются чем-то сложным, потому что принимают во внимание структуру ресурса. Каждый, кто хочет понять какой-нибудь ресурс, должен понять его API - будь то JDBC и SQL для реляционных БД или W3C или JDOM для XML. Это делает ПО не только менее понятным, но еще это делает изменения гораздо более сложными, например, если вы собираетесь перейти со временем с SQL на XML.

Решением здесь является обертывание всего специального API в класс, интерфейс которого выглядит как интерфейс обычного объекта. Остальные объекты обращаются к ресурсу через этот Шлюз, который транслирует эти простые вызовы в соответствующий специальный API-код.

6.4 Шлюз к ClickHouse

Диаграмма классов шлюза представлена на рисунке 11.

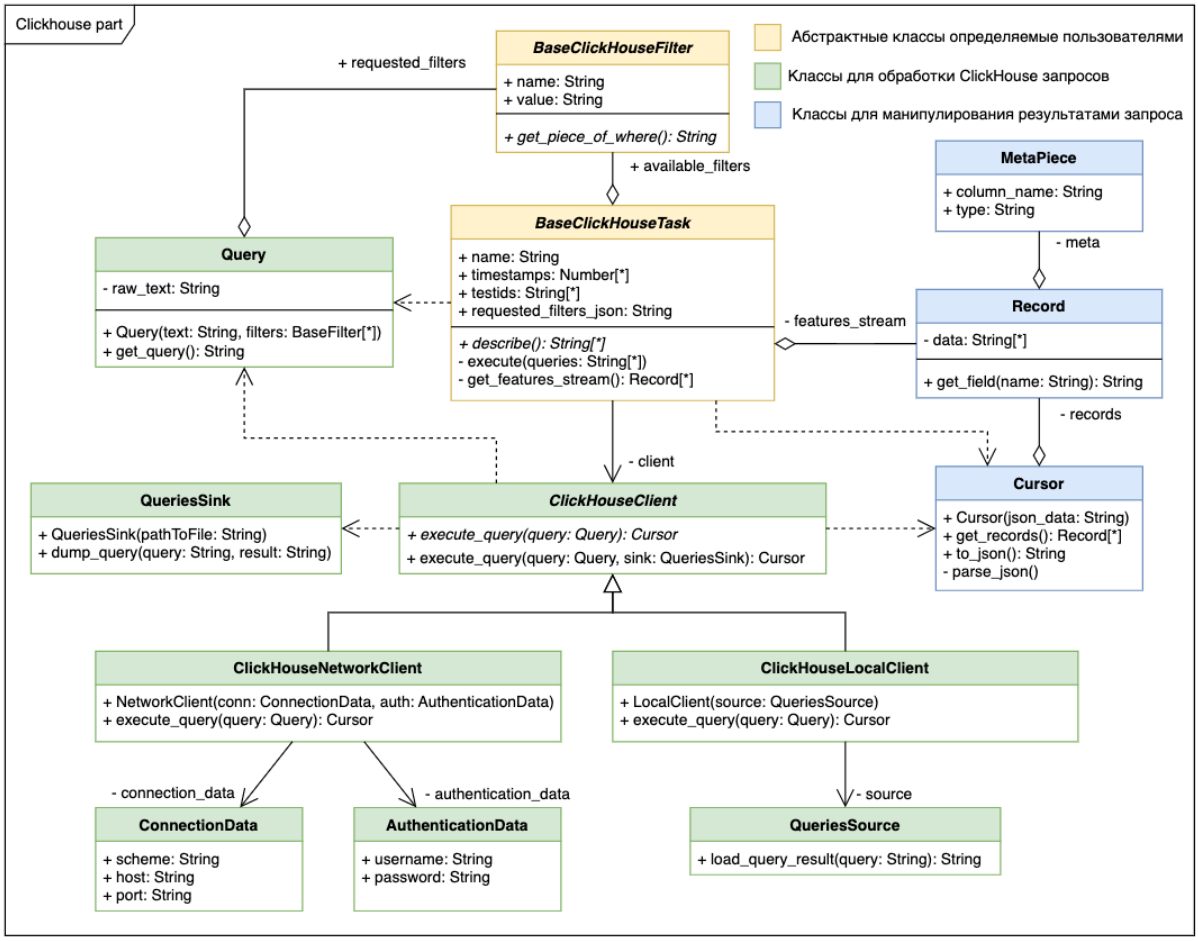


Рисунок 11 – Шлюз к ClickHouse

Рассмотрим более детально шлюз к ClickHouse. Начнем с классов для обработки ClickHouse-запросов:

- Query – это класс, который в конструкторе принимает SQL-запрос и список фильтров, которые нужно применить к запросу. Данный класс имеет метод get_query, который возвращает текст конечного запроса.
- ClickHouseClient – это абстрактный клиент к ClickHouse. Имеет две перегрузки одного метода execute_query. Метод с одним аргументом принимает на вход запрос, обрабатывает его и возвращает результат. Метод с двумя аргументами делает то же самое, что и метод с одним

аргументом, но он дополнительно записывает результаты запроса в QueriesSink.

- ClickHouseNetworkClient – это одна из реализаций ClickHouseClient. Данная реализация нужна для взаимодействия с СУБД по сети.
- ConnectionData и AuthenticationData – это классы, содержащие информацию, необходимые для подключения к ClickHouse по сети. Данные классы используются в ClickHouseNetworkClient.
- ClickHouseLocalClient – это реализация ClickHouseClient, используемая для написания тестов.
- QuriesSink – это класс, необходимый для записи результатов запроса к ClickHouse в файл. Это может понадобиться для локальной отладки или написания тестов.
- QureisSouce – это класс, который по тексту запроса выдает ответ. Может быть использован для локальной отладки или при написании тестов.

Теперь перейдем к рассмотрению классов для манипулирования результатами запроса:

- MetaPiece – это кусочек метаданных таблицы. Содержит в себе название колонки и тип данных в ней.
- Record – это одна строка из результата запроса. Данный класс позволяет по названию поля получать его значение.
- Cursor – это класс, позволяющий итерироваться по результату запроса. В его задачи входит распарсить json от ClickHouse и предоставить метод get_records, который возвращает список Record.

Вернемся к рассмотрению классов, которые доступны для наследования – BaseClickHouseTask и BaseClickHouseFilter.

- BaseClickHouseTask – это базовый абстрактный класс расчета в ClickHouse. При создании конкретного расчета аналитик создает свой класс на основе BaseClickHouseTask. Внутри базового класса

содержатся все необходимые поля для запуска расчета. Это testids – идентификаторы выборок из A/B-тестирования, timestamps – даты, за которые необходимо произвести расчет, requested_filters_json – запрошенные фильтры, которые задают определенный срез пользователей. Метод, в котором описываются SQL-запросы называется describe, ожидается, что этот метод вернет список SQL-запросов. Метод execute умеет запускать список запросов, полученных из метода describe. Оба вышеперечисленных метода вызываются внутри метода get_features_stream, который определен в BaseTask. Метод get_features_stream возвращает список записей, состоящий из названий и значений свойств. Идея заключается в том, что нам не важно, из какого хранилища данных были получены свойства и их значения, именно поэтому метод get_features_stream объявлен в BaseTask и обязателен в реализации в BaseYtTask и BaseClickHouseTask.

- BaseClickHouseFilter – это класс, который предоставляет возможность создавать фильтры, которые нужны для задания срезов пользователей. У каждого фильтра есть name – название и value – значение фильтра. Например, название фильтра – platform_filter, а значение фильтра – desktop. Также в фильтре должен быть реализован метод get_piece_of_where, именно в этом методе описано, как применить фильтр к логам, ожидается, что в результате этого метода будет строка, которую можно вставить в WHERE часть SQL-запроса.

6.5 Консольная утилита

Конечным артефактом разработки системы обработки результатов экспериментов A/B-тестирования является консольная утилита. Операционная система, в которой работает консольная утилита – Linux.

Linux – семейство UNIX-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты [21].

Рассмотрим пример запуска консольной утилиты и основные параметры запуска:

./calc_module fetch --project alice --task main --dates 20220230 --config /path/to/config

- calc_module – имя исполняемого файла, а именно результат компиляции исходных файлов системы обработки результатов экспериментов A/B-тестирования.
- fetch – тип запуска, означающий, что мы хотим получить метрики. Для расчетов в ClickHouse актуален только такой тип запуска. Для расчетов в YT есть тип запуска collect, который означает, что мы хотим сделать предрасчет логов в свойства.
- --project – название проекта, которое было указано в поле класса, унаследованном от BaseApp.
- --task – название расчета, которое было указано в поле класса, унаследованном от BaseYtTask или BaseClickHouseTask.
- --dates – даты, за которые мы хотим произвести расчет. Можно указать одну дату или интервал из дат.
- --config – json-файл, конфигурирующий расчет. В этом файле указывается, какие эксперименты на каких срезах нужно рассчитать. Пример конфигурационного файла представлен на рисунке 12.

```
[
  {
    "testids": [435, 436],
    "filters": [{"name": "platform", "value": "app"}]
  }
]
```

Рисунок 12 - пример конфигурационного файла

Рассмотрим конфигурационный файл из рисунка 12 подробнее. В нем указано, что мы хотим посчитать один срез, состоящий из двух выборок с идентификаторами 435 и 436. Данный срез имеет однин фильтр, имеющий название platform и значение app.

В ходе проектирования шлюза к ClickHouse в системе обработки результатов экспериментов A/B-тестирования интерфейс консольной утилиты не изменился. Этого удалось достичь с помощью BaseTask, общего базового класса для BaseYtTask и BaseClickHouseTask.

ЗАКЛЮЧЕНИЕ

В рамках данной работы был выполнен анализ требований, составлена модель предметной области. Были выявлены проблемы в текущем решение, рассмотрены способы решения описанных проблем. В качестве решения поставленных проблем была выбрана СУБД ClickHouse. На основе полученной модели предметной области был спроектирован и реализован шлюз к ClickHouse. Все поставленные цели и задачи были достигнуты в соответствии со сформулированными требованиями.

В момент написания данной работы 4 проекта используют ClickHouse в своих расчетах, это 8% от общего числа проектов. При этом было добавлено 30 расчетов, написанных к ClickHouse, что составляет 15% от общего количества расчетов.

В результате можно сказать, что внедрение шлюза к ClickHouse было проведено успешно, а данная функциональность нашла своих пользователей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Google Analytics Experiments [Электронный ресурс]: Google аналитика – URL: https://developers.google.com/analytics/devguides/collection/analyticsjs/experiments?hl=ru (дата обращения: 20.05.2022).
2. MapReduce [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/MapReduce (дата обращения: 20.05.2022).
3. Apache Hadoop [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/Hadoop (дата обращения: 20.05.2022).
4. Apache Spark [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/Apache_Spark (дата обращения: 20.05.2022).
5. Yandex Cloud [Электронный ресурс]: Официальный сайт Yandex Cloud – URL: https://console.cloud.yandex.ru (дата обращения: 20.05.2022).
6. Redis [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/Redis (дата обращения: 20.05.2022).
7. ClickHouse [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/ClickHouse (дата обращения: 20.05.2022).
8. PostgreSQL [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/PostgreSQL (дата обращения: 20.05.2022).
9. MySQL [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/MySQL (дата обращения: 20.05.2022).
10. MongoDB [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/MongoDB (дата обращения: 20.05.2022).

11. Greenplum [Электронный ресурс]: Wikipedia, the free encyclopedia – URL: https://en.wikipedia.org/wiki/Greenplum (дата обращения: 20.05.2022).
12. Типы данных PostgreSQL [Электронный ресурс]: Документация к PostgreSQL – URL: https://postgrespro.ru/docs/postgresql/9.6/datatype (дата обращения: 20.05.2022).
13. Тест производительности ClickHouse в сравнение с другими СУБД [Электронный ресурс]: Официальный сайт ClickHouse – URL: https://clickhouse.com/benchmark/dbms/#[100000000,[%22ClickHouse%22,%22MySQL%22,%22Greenplum%22,%22PostgreSQL%22],[%220%22]] (дата обращения: 20.05.2022).
14. Python [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/Python (дата обращения: 20.05.2022).
15. C++ [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/C%2B%2B (дата обращения: 20.05.2022).
16. Requests [Электронный ресурс]: Wikipedia, the free encyclopedia – URL: https://en.wikipedia.org/wiki/Requests_(software) (дата обращения: 20.05.2022).
17. Json в Python [Электронный ресурс]: Документация к Python – https://docs.python.org/3/library/json.html (дата обращения: 20.05.2022).
18. Что такое ClickHouse [Электронный ресурс]: Документация к ClickHouse – URL: https://clickhouse.com/docs/ru (дата обращения: 20.05.2022).
19. Форматы входных и выходных данных в ClickHouse [Электронный ресурс]: Документация к ClickHouse – URL: https://clickhouse.com/docs/en/interfaces/formats/ (дата обращения: 20.05.2022).

20. Паттерн шлюз [Электронный ресурс]: Сайт Мартина Фаулера – URL: https://martinfowler.com/articles/gateway-pattern.html (дата обращения: 20.05.2022).
21. Linux [Электронный ресурс]: Википедия – свободная энциклопедия – URL: https://ru.wikipedia.org/wiki/Linux (дата обращения: 20.05.2022).

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: юдаков алексей александрович
Проверяющий: Юдаков Алексей (gigomaster@yandex.ru / ID: 6212091)

Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

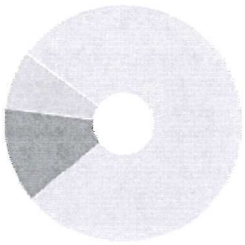
С результатом ознакомлен Моисеев А.М.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 3
Начало загрузки: 25.05.2022 13:20:42
Длительность загрузки: 00:00:04
Имя исходного файла: ВКР Юдаков.pdf
Название документа: ВКР Юдаков
Размер текста: 1 кБ
Символов в тексте: 50461
Слов в тексте: 6122
Число предложений: 377

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 25.05.2022 13:20:47
Длительность проверки: 00:01:23
Комментарии: не указано
Поиск с учетом редактирования: да
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по eLIBRARY.RU (KkRu), Переводные заимствования по eLIBRARY.RU (KyRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования по Интернету (KkRu), Переводные заимствования по Интернету (KyRu), Переводные заимствования (KkEn), Переводные заимствования (KyEn), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, СПС ГАРАНТ, Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
13,23%	0%	7,74%	79,03%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте	Комментарии
[01]	6,51%	6,51%	не указано	13 Янв 2022	Библиография	1	1	
[02]	0,19%	4,95%	Обзор - Документация ClickHouse https://clickhouse.tech	20 Мая 2020	Интернет Плюс	10	23	
[03]	0%	4,95%	Обзор - Документация ClickHouse https://clickhouse.tech	20 Мая 2020	Интернет Плюс	0	23	
[04]	0,6%	4,76%	Обзор | Документация ClickHouse https://clickhouse.tech	09 Июн 2021	Интернет Плюс	3	8	
[05]	0%	4,18%	ClickHouse — Национальная библиотека им. Н. Э. Баумана https://ru.bmstu.wiki	30 Мая 2021	Интернет Плюс	0	6	
[06]	0%	4,18%	ClickHouse — Национальная библиотека им. Н. Э. Баумана https://ru.bmstu.wiki	05 Дек 2021	Интернет Плюс	0	6	
[07]	0%	4,1%	Что такое ClickHouse | ClickHouse Docs https://clickhouse.com	25 Мая 2022	Интернет Плюс	0	4	
[08]	0%	3,83%	Классен, Роман Константинович Консервативные СУБД класса BigData с регулярным планом обработки запросов на кластерной платформе : диссертация ... кандидата технических наук : 05.13.11 Казань 2019 http://dlib.rsl.ru	01 Янв 2019	Сводная коллекция РГБ	0	6	
[09]	0%	3,17%	"Научно-аналитический журнал ""Инновации и инвестиции"" №2/2017" https://book.ru	21 Янв 2020	Сводная коллекция ЭБС	0	6	
			Материалы подсекции "Программная					

[10]	0,02%	2,94%	инженерия" на конференции "Наука и Молодежь-2019" https://altstu.ru	19 Июл 2019	Интернет Плюс	1	17
[11]	0%	2,92%	247764 http://e.lanbook.com	10 Мар 2016	Сводная коллекция ЭБС	0	5
[12]	0,12%	2,71%	Форшев Даниил Форшев_ВКР_2018.docx	14 Июн 2018	Кольцо вузов	1	2
[13]	0%	2,71%	Разработка системы хранения и анализа данных промышленной телеметрии	25 Июн 2018	Кольцо вузов	0	2
[14]	0%	2,71%	Системный администратор: ежемесячный журнал. 2017. № 3(172) http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	2
[15]	0,16%	2,71%	От экспертов 1С-Рарус: Как и зачем интегрировать Yandex ClickHouse с 1С? https://rarus.ru	18 Апр 2021	Интернет Плюс	1	9
[16]	0%	2,71%	От экспертов 1С-Рарус: Как и зачем интегрировать Yandex ClickHouse с 1С? https://rarus.ru	25 Мая 2022	Интернет Плюс	0	9
[17]	0%	2,7%	ClickHouse в системах сбора статистики. http://elibrary.ru	04 Мая 2017	eLIBRARY.RU	0	4
[18]	0%	2,67%	Проблемы регулирования модификации программного обеспечения. http://elibrary.ru	23 Сен 2020	eLIBRARY.RU	0	4
[19]	0%	2,61%	Андреев, Михаил Владимирович Управление сетевым взаимодействием в цепях поставок научно-производственных предприятий : диссертация ... кандидата технических наук : 05.13.10 Самара 2015 http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	0	5
[20]	2,58%	2,58%	ClickHouse в системах сбора статистики. http://elibrary.ru	04 Мая 2017	Перефразирования по eLIBRARY.RU	1	1
[21]	2,36%	2,36%	ВКР_Коровин_КН.pdf	23 Июн 2020	Кольцо вузов	2	2
[22]	0%	2,31%	http://kek.fknt.donntu.org/sites/default/files/becs_2019.pdf http://kek.fknt.donntu.org	08 Янв 2022	Интернет Плюс	0	12
[23]	0%	2,27%	Деривационное поле локативности в аспекте межсистемного сопоставления (русский литературный язык / говоры) http://dep.nlb.by	11 Ноя 2016	Диссертации НББ	0	5
[24]	0%	2,27%	2020_ИЭИТУС_ПОВТАС_09_04_04_МД_Рязанов_Олег_Юрьевич	25 Июн 2020	Кольцо вузов	0	1
[25]	0%	2,18%	Пространства городской цивилизации: идеи, проблемы, концепции : материалы Международной научной конференции (4-5 октября, 2017 г.) http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	4
[26]	0%	2,18%	Переломы проксимального отдела бедренной кости у взрослых: объективизация причин летальных исходов в катамнезе http://dep.nlb.by	16 Янв 2020	Диссертации НББ	0	4
[27]	0%	2,18%	Стимулирование товарного экспорта в условиях экономической интеграции http://dep.nlb.by	16 Янв 2020	Диссертации НББ	0	4
[28]	1,84%	2,14%	Tsarkov_VKR	14 Июн 2021	Кольцо вузов	1	2
[29]	0%	1,96%	ИНТЕРПРЕТАТОР ИСПОЛНИТЕЛЯ «РОБОТ». http://elibrary.ru	04 Авг 2016	eLIBRARY.RU	0	7
[30]	0%	1,96%	ClickHouse в системах сбора статистики	08 Янв 2019	СМИ России и СНГ	0	10
[31]	0%	1,82%	ДипломнаяработаЛУИ	02 Июн 2021	Кольцо вузов	0	1
[32]	0%	1,82%	Выпускная квалификационная работа_Турышев Д.А..docx	16 Июн 2021	Кольцо вузов	0	1
[33]	0%	1,78%	borohov_a_s_optimizaciya-processa-obrabotki-zayavok-na-podkreplenie-momentalnyh-kreditnyh-kart-s-primeneniem-alg.docx	27 Мая 2021	Кольцо вузов	0	1
[34]	0%	1,78%	Pyatova_kursovaya (2).docx	17 Мая 2021	Кольцо вузов	0	1
[35]	0%	1,78%	A1	14 Июн 2021	Кольцо вузов	0	1
[36]	0%	1,71%	Курсовая работа по дисциплине Алгоритмы и структуры данных https://topuch.ru	17 Мая 2022	Интернет Плюс	0	4

[37]	0%	1,54%	Почему инвестиции — это еще одно ярмо на шею рабочему человеку? http://imperiyanews.ru	21 Мая 2020	СМИ России и СНГ	0	11
[38]	0%	1,45%	Услуги как объект гражданских прав. Дипломная (ВКР). Антикризисный менеджмент. 2015-03-27 https://bibliofond.ru	15 Янв 2021	Интернет Плюс	0	20
[39]	0%	1,4%	Скачать http://sworld.com.ua	27 Ноя 2016	Интернет Плюс	0	13
[40]	0,17%	1,34%	Эволюция структур данных в Яндекс.Метрике / Блог компании Яндекс / Хабр https://habr.com	21 Дек 2019	Интернет Плюс	2	7
[41]	0%	1,33%	Слезин, Кирилл Анатольевич Аналитические и процедурные модели интеллектуальной геоинформационной системы визуализации контуров лесных пожаров : диссертация ... кандидата технических наук : 05.25.05 Тамбов 2018 http://dlib.rsl.ru	14 Июн 2019	Сводная коллекция РГБ	0	3
[42]	0%	1,33%	ФАКТОРЫ ДИНАМИКИ РОЖДАЕМОСТИ НАСЕЛЕНИЯ РОССИИ В НАЧАЛЕ XXI ВЕКА	31 Дек 2018	СМИ России и СНГ	0	11
[43]	1,23%	1,23%	не указано	13 Янв 2022	Шаблонные фразы	15	15
[44]	1,19%	1,19%	patterns of enterprise appl.rar/Patterns of Enterprise Appl.pdf http://inethub.olvi.net.ua	09 Янв 2018	Переводные заимствования (RuEn)	1	1
[45]	0%	1,12%	Эволюция структур данных в Яндекс.Метрике http://pcnews.ru	08 Янв 2019	СМИ России и СНГ	0	3
[46]	0,07%	1,1%	Download http://elib.spbstu.ru	07 Сен 2019	Интернет Плюс	1	13
[47]	0%	0,9%	Колибри - Разработка https://octonica.ru	30 Ноя 2020	Интернет Плюс	0	5
[48]	0%	0,89%	Информационно-управляющие системы: научный журнал. 2014. № 2(69) http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1
[49]	0,06%	0,86%	Microsoft - Юнионпедия https://ru.unionpedia.org	25 Ноя 2019	Интернет Плюс	2	4
[50]	0%	0,7%	ПП 2017 г.н. http://asu.ru	11 Мая 2020	Интернет Плюс	0	6
[51]	0,21%	0,68%	ИНТЕРПРЕТАТОР ИСПОЛНИТЕЛЯ «РОБОТ». http://elibrary.ru	04 Авг 2016	Перефразирования по eLIBRARY.RU	1	1
[52]	0%	0,64%	https://shelly.kpfu.ru/e-ksu/docs/DISSERTATION/F_10188008/Dissertaciya_Klassen_RK.pdf https://shelly.kpfu.ru	29 Окт 2019	Интернет Плюс	0	6
[53]	0%	0,6%	https://unecon.ru/sites/default/files/dissgrishinael.pdf https://unecon.ru	24 Фев 2022	Интернет Плюс	0	3
[54]	0,59%	0,59%	A/B тест — это просто / Хабрахабр http://habrahabr.ru	01 Янв 2017	Перефразирования по Интернету	1	1
[55]	0%	0,59%	patterns of enterprise appl.rar/Patterns of Enterprise Appl.pdf http://inethub.olvi.net.ua	09 Янв 2018	Переводные заимствования (RuEn)	0	1
[56]	0,15%	0,56%	РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ СБОРА, АНАЛИЗА И ХРАНЕНИЯ ДАННЫХ О БРОНИРОВАНИИ И ЗАКАЗАХ ДЛЯ МАЛЫХ ПРЕДПРИЯТИЙ, РАБОТАЮЩИХ В СФЕРЕ ОКАЗАНИЯ УСЛУГ. http://elibrary.ru	15 Окт 2019	eLIBRARY.RU	1	2
[57]	0,53%	0,53%	Опубликован релиз СУБД Redis 7.0 http://tadviser.ru	29 Апр 2022	СМИ России и СНГ	1	1
[58]	0%	0,47%	Сравнение СУБД для ГИС - Системы управления базами данных, применяемы в геоинформационных системах https://studwood.ru	03 Апр 2021	Интернет Плюс	0	4
[59]	0%	0,46%	Использование Clickhouse в качестве замены ELK, Big Query и TimescaleDB / Блог компании ua-hosting.company / Хабр https://habr.com	23 Мая 2022	Интернет Плюс	0	4
			Разработка модели на основе теории массового обслуживания для				

[60]	0,36%	0,45%	исследования загрузки сети зарядных станций http://elib2.altstu.ru	08 Сен 2017	Интернет Плюс	2	3
[61]	0,45%	0,45%	Сравнение систем анализа Big Data. http://elibrary.ru	27 Мая 2019	eLIBRARY.RU	1	1
[62]	0%	0,44%	ТОП 50 лучших программ для создания аналитических отчетов бизнесу - xmldatafeed.com https://xmldatafeed.com	18 Дек 2020	Интернет Плюс	0	1
[63]	0%	0,44%	ТОП 50 лучших программ для создания аналитических отчетов бизнесу - xmldatafeed.com https://xmldatafeed.com	10 Мая 2022	Интернет Плюс	0	1
[64]	0%	0,41%	ClickHouse http://ru.wikipedia.org	10 Сен 2019	Интернет Плюс	0	1
[65]	0%	0,41%	ClickHouse http://ru.wikipedia.org	02 Сен 2020	Интернет Плюс	0	1
[66]	0%	0,41%	LogiCH - хранение и анализ журнала регистрации в сверхбыстрой СУБД ClickHouse https://infostart.ru	25 Мая 2022	Интернет Плюс	0	1
[67]	0%	0,41%	Как сменить профессию и стать крутым аналитиком? / Хабр https://habr.com	25 Мая 2022	Интернет Плюс	0	1
[68]	0,35%	0,41%	СУБД «Яндекс ClickHouse» внедрили в российскую систему предиктивной аналитики «ПРАНА» https://habr.com	08 Фев 2021	СМИ России и СНГ	2	1
[69]	0,12%	0,4%	Базовые технологии разработки сайтов и сопутствующего программного обеспечения (середина октября 2019 года, с упором на использования 1С-битрикс) https://moscowwebstudio.ru	13 Окт 2020	Интернет Плюс	1	2
[70]	0%	0,4%	Материалы Юбилейной студенческой научно-практической конференции экономического факультета ТГУ, посвященной 50-летию факультета. Томск, 19-20 апреля 2013 г // Труды Томского государственного университета. 2013. Т. 287 http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1
[71]	0%	0,39%	274102 http://biblioclub.ru	20 Апр 2016	Сводная коллекция ЭБС	0	1
[72]	0%	0,39%	62154 http://e.lanbook.com	09 Мар 2016	Сводная коллекция ЭБС	0	1
[73]	0%	0,39%	MapReduce http://ru.wikipedia.org	10 Сен 2019	Интернет Плюс	0	1
[74]	0%	0,39%	https://e-mba.ru/uploads/campus/54c1948d-51b2-40f7-a618-9abe9232e7b6/big_data_i_machine_learning.pdf https://e-mba.ru	25 Мая 2022	Интернет Плюс	0	1
[75]	0%	0,39%	Сложные геоинформационные системы. http://elibrary.ru	11 Фев 2020	eLIBRARY.RU	0	1
[76]	0%	0,39%	Обработка больших данных: первые шаги в понимании Hadoop MapReduce и Spark https://pcnews.ru	23 Июл 2021	СМИ России и СНГ	0	1
[77]	0%	0,33%	Обработка и передача учетных данных для классических и цифровых электроподстанций http://studentlibrary.ru	20 Дек 2016	Медицина	0	1
[78]	0,16%	0,33%	https://www.rsu.edu.ru/wp-content/uploads/opop19/rpd014/%D0%911.%D0%91.7.%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%BE-%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D0%B2%D1%8B%D0%B5_%D0%B0%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D1%8B_%D1%80%D1%8B%D0%BD%D0%BA%D0%B0_%D... https://rsu.edu.ru	04 Фев 2022	Интернет Плюс	2	2
[79]	0,32%	0,32%	https://old.kai.ru/science/disser/files/file_310/text_diss.pdf https://old.kai.ru	09 Апр 2022	Интернет Плюс	1	1
[80]	0%	0,31%	Разработка проекта Web-приложения для бронирования столиков и мест в кафе https://revolution.allbest.ru	24 Мая 2022	Интернет Плюс	0	1

[81]	0%	0,31%	Linux - Юнионпедия https://ru.unionpedia.org	30 Мая 2021	Интернет Плюс	0	1	
[82]	0%	0,27%	РАСЧЕТ ИНФОРМАЦИОННЫХ РИСКОВ БЕЗОПАСНОСТИ ВЕБ-ПРИЛОЖЕНИЯ. http://elibrary.ru	09 Июл 2020	eLIBRARY.RU	0	1	
[83]	0,27%	0,27%	https://nauchkor.ru/uploads/documents/5f11a8c7cd3d3e0001acba3f.pdf https://nauchkor.ru	25 Мая 2022	Интернет Плюс	1	1	
[84]	0%	0,27%	https://lib.tsu.ru/win/produkzija/metodichka/pril1.pdf https://lib.tsu.ru	25 Мая 2022	Интернет Плюс	0	1	
[85]	0%	0,26%	A/B тест — это просто http://habrahabr.ru	04 Янв 2019	СМИ России и СНГ	0	1	
[86]	0,26%	0,26%	Высшая математика в примерах и задачах. Т. 3 http://studentlibrary.ru	19 Дек 2016	Медицина	1	1	
[87]	0%	0,25%	Вагнер, Дмитрий Викторович Высокочастотные электромагнитные характеристики композиционных радиоматериалов на основе гексагональных ферритов : диссертация ... кандидата технических наук : 01.04.03 Томск 2019 http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	0	1	
[88]	0%	0,25%	Разработка информационного сайта для проекта «Живая История»: выпускная квалификационная работа http://biblioclub.ru	21 Янв 2020	Сводная коллекция ЭБС	0	1	
[89]	0%	0,25%	Павлов, Дмитрий Александрович Автоматизированное управление технологическими процессами с использованием алгоритмов нечеткой фильтрации : диссертация ... кандидата технических наук : 05.13.06 смоленск 2015 http://dlib.rsl.ru	22 Авг 2019	Сводная коллекция РГБ	0	1	
[90]	0%	0,25%	Методы решения обратных задач с использованием фильтра Калмана. http://elibrary.ru	17 Окт 2015	eLIBRARY.RU	0	1	
[91]	0%	0,24%	АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДЛЯ АДМИНИСТРАТИВНОГО КОНТРОЛЯ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ. http://elibrary.ru	15 Окт 2019	eLIBRARY.RU	0	1	
[92]	0,08%	0,21%	Устройство системы непрерывной интеграции в Яндексе http://pcnews.ru	28 Ноя 2018	СМИ России и СНГ	2	2	
[93]	0%	0,21%	https://dspace.tltsu.ru/bitstream/123456789/4119/1/%D0%A8%D0%B0%D0%BD%D0%B4%D1%83%D1%80%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%90.%D0%92._%D0%9F%D0%98%D0%B1-1301.pdf https://dspace.tltsu.ru	23 Апр 2022	Интернет Плюс	0	2	
[94]	0%	0,16%	История языка программирования Python - InterestPrograms.RU https://interestprograms.ru	23 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[95]	0%	0,11%	Анализ средств и моделей взаимодействия между компонентами в системе управления корпоративной мобильностью	20 Дек 2018	СМИ России и СНГ	0	1	Источник исключен. Причина: Маленький процент пересечения.